UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/664,941 | 09/19/2000 | Kuo-Chun Lee | CREO.005US0 | 5234 |

25242 7590 10/08/2004

VICTOR H. OKUMOTO
P.O. BOX 6120
FREMONT, CA 94538

| EXAMINER | |
|---|---|
| MASKULINSKI, MICHAEL C | |
| **ART UNIT** | **PAPER NUMBER** |
| 2113 | |

DATE MAILED: 10/08/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) |
|---|---|---|
| | 09/664,941 | LEE ET AL. |
| | Examiner | Art Unit |
| | Michael C Maskulinski | 2113 |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *20 July 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-37* is/are pending in the application.

4a) Of the above claim(s) *4,14-19,21 and 32-37* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3,5-13,20 and 22-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

**Final Office Action**

***Claim Rejections - 35 USC § 102***

1. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

2. Claims 1-3, 5, 6, 9-11, 22-24, and 28-30 are rejected under 35 U.S.C. 102(e) as being anticipated by Fuh et al., U.S. Patent 6,324,683 B1.

Referring to claims 1 and 9:

a. In column 12, lines 8-30, Fuh et al. disclose that the debugger is initiated by the client (detecting a debug request initiated by a user of a client computer to debug an application program on said client computer).

b. In column 33, lines 40-47, Fuh et al. disclose that the debugger client and debugger server can communicate with the tool locator through a socket connection such as a connectionless internet family socket that is bound to an internet address specified in an environment variable named TOOLLOCATORHOST, and a reserved well-known port (transmitting said debug request to a server computer over the Internet).

c. In column 11, lines 60-62, Fuh et al. disclose that each connected client is serviced by a database engine that is running in a set of threads on the database server machine within a process that is shared by other clients (establishing a connection between said client computer and said server computer over the Internet).

d-g. In column 12, lines 8-30, Fuh et al. teach receiving a request from a debug program of the server computer; causing an application program of the client computer to generate a response to the request; and transmitting an indication of the response back to the debug program; and repeating these steps multiple times so as to run the application program through a diagnostic sequence.

Referring to claims 2, 10, 23, and 29, in column 10, lines 13-15, Fuh et al. disclose that the debugger issues a series of RPC calls to obtain a current context of the underlying external program and displays the current context state along with a frozen external program state to a user. The user can then use all of the debugging functions of the debugger on the external program (wherein said diagnostic sequence is provided to said debug program by a user of said server computer).

Referring to claims 3, 11, 24, and 30, in column 10, lines 4-13, Fuh et al. disclose that execution of the external program is suspended while the debugger executes a remote procedure call (RPC) to get the most recent invocation stack frame record from the DBMS. The debugger then sets a breakpoint at the entry point of the external program. When the external program is executed, the breakpoint set by the debugger is encountered (wherein said diagnostic sequence is preprogrammed into said debug program).

Referring to claim 5, in column 56, lines 53-67, Fuh et al. disclose that the user needs to specify the various programs that make up the application. For each program the user needs to specify the program arguments, the host machine the program should

run on, and the user ID the program should run under (transmitting identifications of said application program and said client computer to said server computer).

Referring to claim 6, in column 56, lines 53-67, Fuh et al. disclose that the user needs to specify the various programs that make up the application. For each program the user needs to specify the program arguments, the host machine the program should run on, and the user ID the program should run under. Further, optionally, additional information may be specified for passwords, debugging options, etc. (transmitting a user identification and a password provided by a user of said client computer to said server computer).

Referring to claims 22 and 28:

a. In column 12, lines 8-30, Fuh et al. disclose that the debugger is initiated by the client. Further, in column 33, lines 40-47, Fuh et al. disclose that the debugger client and debugger server can communicate with the tool locator through a socket connection such as a connectionless internet family socket that is bound to an internet address specified in an environment variable named TOOLLOCATORHOST, and a reserved well-known port (receiving a request from a user of a client computer over the Internet to debug an application program of said client computer).

b-d. In column 12, lines 8-30, Fuh et al. teach transmitting back to said client computer a request for said application program to take an action; receiving an indication of a response of said application program action back from said client

computer; and repeating the steps multiple times so as to run the application program through a diagnostic sequence.

***Claim Rejections - 35 USC § 103***

3. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

4. Claims 7, 8, 12, 13, 20, 27, and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Fuh et al., U.S. Patent 6,324,683 B1 and further in view of Schauser, U.S. Patent 6,331,855 B1.

Referring to claims 7 and 12:

h. In column 10, lines 4-15, Fuh et al. disclose that when the external program is executed, the breakpoint set by the debugger is encountered. The debugger then issues a series of RPC calls to obtain a current context of the underlying external program and displays the current context state along with a frozen external program state to a user. The user can then use all of the debugging functions of the debugger on the external program (the preprogrammed diagnostic sequence is paused by a user of said server computer and control of said debug program is transferred to said user of said server computer and receiving a request initiated by said user of said server computer).

i. In column 10, lines 4-15, Fuh et al. disclose that the user can then use all of the debugging functions of the debugger on the external program (causing said application program to respond to said request).

j., k. In column 10, lines 4-15, Fun et al. disclose displaying the current context state along with a frozen external program state to a user. However, Fuh et al. don't explicitly disclose generating a graphics file including pixel information for a graphics image displayed on a display screen of said client computer and automatically transmitting said graphics file to said server computer so that said graphics image is displayed on a display screen of said server computer. In column 2, lines 31-44, Schauser discloses that the present invention is a system and method for controlling information displayed on a first processor-based system. The system comprises a memory to store instruction sequences by which the second processor-based system is processed, and a processor coupled to the memory. The stored instruction sequences cause the processor to: (a) examine, at predetermined interval, a location of a currently displayed image; (b) compare the location with a corresponding location of a previously displayed image to determine if the previously displayed image has changed; (c) transmitting location information representing the change; and (d) storing the changed information on the first processor-based system. It would have been obvious to one of ordinary skill at the time of the invention to include the display device of Schauser into the system of Fuh et al. A person of ordinary skill in the art would have been motivated to make the modification because *remote desktop*

*access technology allows a user to control a remote computer as if sitting right in front of it. The user can run applications, access files, change configurations, or debug problems. There are many different uses for such technology, including providing technical support, telecommuting, collaboration, education and training, equipment control, software and computer rental, software demonstration, sales presentations, and access from mobile handheld devices* (see Schauser: column 1, lines 15-23).

Referring to claims 8 and 13, in column 10, lines 4-15, Fuh et al. teach repeating the steps multiple times so as to run said application program through a diagnostic sequence.

Referring to claim 20:

a. In column 12, lines 8-30, Fuh et al. disclose that the debugger is initiated by the client (detecting a debug request initiated by a user of a client computer).

b. In column 33, lines 40-47, Fuh et al. disclose that the debugger client and debugger server can communicate with the tool locator through a socket connection such as a connectionless internet family socket that is bound to an internet address specified in an environment variable named TOOLLOCATORHOST, and a reserved well-known port. In column 11, lines 60-62, Fuh et al. disclose that each connected client is serviced by a database engine that is running in a set of threads on the database server machine within a process that is shared by other clients (establishing a connection between said client computer and said server computer over the Internet).

c., d. In column 12, lines 8-30, Fuh et al. teach receiving a request from a debug program of said server computer and causing an application program of said client computer to respond to said request.

e., f. In column 10, lines 4-15, Fun et al. disclose displaying the current context state along with a frozen external program state to a user. However, Fuh et al. don't explicitly disclose generating a graphics file including pixel information for a graphics image displayed on a display screen of said client computer and automatically transmitting said graphics file to said server computer so that said graphics image is displayed on a display screen of said server computer. In column 2, lines 31-44, Schauser discloses that the present invention is a system and method for controlling information displayed on a first processor-based system. The system comprises a memory to store instruction sequences by which the second processor-based system is processed, and a processor coupled to the memory. The stored instruction sequences cause the processor to: (a) examine, at predetermined interval, a location of a currently displayed image; (b) compare the location with a corresponding location of a previously displayed image to determine if the previously displayed image has changed; (c) transmitting location information representing the change; and (d) storing the changed information on the first processor-based system. It would have been obvious to one of ordinary skill at the time of the invention to include the display device of Schauser into the system of Fuh et al. A person of ordinary skill in the art would have been motivated to make the modification because *remote desktop*

*access technology allows a user to control a remote computer as if sitting right in front of it. The user can run applications, access files, change configurations, or debug problems. There are many different uses for such technology, including providing technical support, telecommuting, collaboration, education and training, equipment control, software and computer rental, software demonstration, sales presentations, and access from mobile handheld devices* (see Schauser: column 1, lines 15-23).

Referring to claims 27 and 31:

e. In column 10, lines 4-15, Fuh et al. disclose that when the external program is executed, the breakpoint set by the debugger is encountered. The debugger then issues a series of RPC calls to obtain a current context of the underlying external program and displays the current context state along with a frozen external program state to a user. The user can then use all of the debugging functions of the debugger on the external program (the preprogrammed diagnostic sequence is paused by a user of said server computer and control of said debug program is transferred to said user of said server computer and receiving a request initiated by said user of said server computer). Further, in column 12, lines 8-30, Fuh et al. teach transmitting to the client computer a request for said application program to take an action.

f-h. In column 10, lines 4-15, Fun et al. disclose displaying the current context state along with a frozen external program state to a user. However, Fuh et al. don't explicitly disclose receiving a graphics file including pixel information for a

graphics image displayed on a display screen of said client computer in response to said action; displaying said graphics image on a display screen of said server computer; and repeating these steps multiple times so as to allow the user of the server computer to interactively debug the application program by transmitting requests for the application program to take certain actions in consideration of graphics images defined in graphics files received from the client computer in response to prior such requests. In column 2, lines 31-44, Schauser discloses that the present invention is a system and method for controlling information displayed on a first processor-based system. The system comprises a memory to store instruction sequences by which the second processor-based system is processed, and a processor coupled to the memory. The stored instruction sequences cause the processor to: (a) examine, at predetermined interval, a location of a currently displayed image; (b) compare the location with a corresponding location of a previously displayed image to determine if the previously displayed image has changed; (c) transmitting location information representing the change; and (d) storing the changed information on the first processor-based system. It would have been obvious to one of ordinary skill at the time of the invention to include the display device of Schauser into the system of Fuh et al. A person of ordinary skill in the art would have been motivated to make the modification because *remote desktop access technology allows a user to control a remote computer as if sitting right in front of it. The user can run applications, access files, change configurations, or debug*

*problems. There are many different uses for such technology, including providing technical support, telecommuting, collaboration, education and training, equipment control, software and computer rental, software demonstration, sales presentations, and access from mobile handheld devices* (see Schauser: column 1, lines 15-23).

5. Claims 25 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fuh et al., U.S. Patent 5,630,049.

Referring to claim 25:

a1 In column 33, lines 48-62, Fuh et al. disclose that a debugger client, can send a message to request debugging services for itself or for another program running on the network. It does this by first sending a message to the tool locator to locate a debugger server specified by the debugger client. The tool locator will return the socket address of a debugger server that matches the debugger client's specification, message. The debugger client then sends a "debugIt" message to the debugger server to request debugging service from the debugger server. (receiving an identification of said application program from said client computer).

a2 In column 33, lines 48-62, Fuh et al. teach receiving information about the application to be debugged. However, Fuh et al. don't explicitly disclose checking said application program identification against an application program identification list to confirm that a contractual obligation exists to debug said application program. The Examiner takes Official Notice that it is well known in

the art of shareware, freeware, and public domain software that the user enters into a contractual agreement with the distributor and receives services based upon this contractual agreement. An example of this is Red Hat Linux. It would have been obvious to one of ordinary skill at the time of the invention to include the concept of a contractual agreement into the system of Fuh et al. A person of ordinary skill in the art would have been motivated to make the modification because checking for a contractual agreement before debugging prevents unwanted users from using the debugging service.

Referring to claim 26:

a3 In column 44, lines 13-20, Fuh et al. disclose a login ID being transmitted to the debugger (transmitting identifications of said client computer to said server computer).

a4 In column 44, lines 13-20, Fuh et al. teach receiving identification of the client computer. However, Fuh et al. don't explicitly disclose confirming that said client computer is authorized to run said application program by comparing said client computer identification against an authorized client computer identification. The Examiner takes Official Notice that it is well known in the art of shareware, freeware, and public domain software that the user enters into a contractual agreement with the distributor and receives services based upon this contractual agreement. An example of this is Red Hat Linux. It would have been obvious to one of ordinary skill at the time of the invention to include the concept of a contractual agreement into the system of Fuh et al. A person of ordinary skill in

the art would have been motivated to make the modification because checking for a contractual agreement before debugging prevents unwanted users from using the debugging service.

***Response to Arguments***

6. Applicant's arguments filed July 20, 2004 have been fully considered but they are not persuasive.

7. On page 12, under the section **Rejection of claims 1-3, 5, 6, 9-11, 22-24, and 28-30 under 35 U.S.C. 102(e)**, the Applicant argues, "In contrast, in Fuh et al., a program such as a user defined function (UDF), stored procedure, or trigger executed in a relational database management system (RDBMS) is to be debugged by a user of a client computer. See, e.g., Abstract. These programs, however, reside on a server computer, not the client computer as in Applicant's case. (emphasis by Applicant) See, e.g., Col. 2, lines 48-51 with regards to a stored procedure on the server computer, Col. 8, lines 1-4 with regards to a UDF on the server computer, and Col. 12, lines 45-46 with regards to the trigger being within the RDBMS, which in turn, is on the server computer (See, e.g., Col. 11, lines 52-55)." The Examiner respectfully disagrees. The Examiner would like to note that in network systems clients and servers are interchangeable. By definition a server is: *On the Internet or other network, a computer or program that responds to commands from a client.*[1] A client by definition is: *On a local area network or the Internet, a computer that accesses shared network resources provided by*

[1] Microsoft Press Computer Dictionary, Third Edition, 1997, page 430.

*another computer.*[2] At any given time, a server could be functioning as a client and a client as a server. In the invention of Fuh et al., the client responds to debug requests from the server and thus takes on the role of a server. Further, in Figure 4, Fuh et al. shows a plurality of servers sharing the debugging engine of the client and thus take on the role of clients.

8. On page 12, under the section **Rejection of claims 1-3, 5, 6, 9-11, 22-24, and 28-30 under 35 U.S.C. 102(e)**, the Applicant argues, "Because the program to be debugged in Fuh et al. resides on the server, it creates three problems-timing, authorization and remote debugging. See, e.g., Col. 7, line 59 to Col. 8, line 29. To eliminate these problems, the program to be debugged initiates the debugger, not the user of the client computer as in Applicant's case. (emphasis by Applicant) Not only does Fuh et al. not teach that the user initiates the debugger, Fuh et al. teaches away (emphasis by Applicant) from such action in order to overcome authorization problems (See, Col. 12, lines 15-19) and remote debugging problems (See, Col. 12, lines 19-25)." The Examiner respectfully disagrees. It is unclear as to how Fuh et al. teach away from a user initiated debugger. In column 12, lines 15-25, Fuh et al. disclose with respect to authorization, the lack of permission to attach UDF process occurs because it is expected that the debugger be initiated by the client. This problem is overcome by granting the debugger such permission since the debugger is initiated from UDF process itself. Likewise, with respect to remote debugging, remote users cannot access the machine where a UDF process runs because it is expected that the debugger be

[2] Microsoft Press Computer Dictionary, Third Edition, 1997, page 92.

initiated by the client. This problem is overcome by redirecting a debugger's standard I/O to the remote machine if the debugger is initiated by a UDF process. Fuh et al. encourage a debugger being initiated from the server (client). Further, in column 56, lines 49-67, Fuh et al. disclose that the debugger initiation is programmed by the user. Thus a user initiates the debugger by programming the server to trigger the client to debug its program.

9. On page 13, under the section **Rejection of claims 1-3, 5, 6, 9-11, 22-24, and 28-30 under 35 U.S.C. 102(e)**, the Applicant argues, "Claim 9 is also believed to be patentable under 35 U.S.C. 102(e) over Fuh et al. since it claims an interface program for 'detecting a debug request initiated by a user of a client computer to debug an application program on said client computer,' and such a task is neither taught nor suggested by Fuh et al. for the reasons stated above in reference to claim 1." The Examiner respectfully disagrees for at least the reasons stated in paragraphs 7 and 8 above.

10. On page 14, under the section **Rejection of claims 1-3, 5, 6, 9-11, 22-24, and 28-30 under 35 U.S.C. 102(e)**, the Applicant argues, "Claim 22 has been amended to claim the task of 'receiving a request from a user of a client computer over the Internet to debug an application program of said client computer,' and such a task is neither taught nor suggested by Fuh et al. for the reasons stated in reference to Claim 1." The Examiner respectfully disagrees for at least the reasons stated in paragraphs 7 and 8 above.

11. On page 14, under the section **Rejection of claims 1-3, 5, 6, 9-11, 22-24, and 28-30 under 35 U.S.C. 102(e)**, the Applicant argues, "Claim 28 has been amended to claim a debug program for 'receiving a request from a user of a client computer over the Internet to debug an application program of said client computer,' and such a debug program is neither taught nor suggested by Fuh et al. for the reasons stated in reference to Claim 1." The Examiner respectfully disagrees for at least the reasons stated in paragraphs 7 and 8 above.

12. On page 15, under the section **Rejection of claims 7, 8, 12, 13, 20, 27, and 31 under 35 U.S.C. 103(a)**, the Applicant argues "Even reversing the roles of the server and client computers does not support a rejection of these claims. For example, in order to combine the teachings of Fuh et al. and Schauser with respect to the transmission of a graphics file, there must be some suggestion in Fuh et al. to do so, and there is no such suggestion. Since Fuh et al. uses X-windows, the user's display (on the client) can be specified as the X-windows server, enabling the UDF to be debugged remotely. See, e.g., Col. 12, lines 27-30. There is no suggestion that it would be desirable to send graphics including pixel information as claimed in Claim 7." The Examiner respectfully disagrees. The test for obviousness is not that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981). Since Fuh et al. already discloses a user interface, any variations would be obvious and

don't necessarily have to be an improvement upon the invention, but rather can be what is available for use by one of ordinary skill in the art.

***Conclusion***

13. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael C Maskulinski whose telephone number is (703)308-6674. The examiner can normally be reached on Monday-Friday 9:30-6:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Robert W Beausoliel can be reached on (703) 305-9713. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MM

ROBERT BEAUSOLIEL
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100